



# UNITED STATES PATENT AND TRADEMARK OFFICE

BS  
UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/657,250	09/06/2000	Ludovic Pierre	5266-02600	9855
44015	7590	10/22/2004	EXAMINER	
OPTV/MEYERTONS THE CHASE BUILDING 700 LAVACA, SUITE 800 AUSTIN, TX 78701				MANNING, JOHN
		ART UNIT		PAPER NUMBER
		2614		

DATE MAILED: 10/22/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	Application No.	Applicant(s)
	09/657,250	PIERRE ET AL.
	Examiner John Manning	Art Unit 2614

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) Responsive to communication(s) filed on 23 July 2004.
- 2a) This action is FINAL.                            2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) Claim(s) 1-29 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) Claim(s) \_\_\_\_\_ is/are allowed.
- 6) Claim(s) 1-25 and 27-29 is/are rejected.
- 7) Claim(s) 26 is/are objected to.
- 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All    b) Some \* c) None of:
  1. Certified copies of the priority documents have been received.
  2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____	5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)
	6) <input type="checkbox"/> Other: _____

## DETAILED ACTION

### ***Response to Arguments***

1. Applicant's arguments with respect to claim 1 haas been considered but are moot in view of the new ground(s) of rejection.

### ***Claim Rejections - 35 USC § 103***

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1-2, 4, 7-10 and 12-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Korn et al. (US Pat No 6,611,877).

In regard to claim 1, Korn et al. discloses an event notification system. The event broker is met by the operating system 22 as shown in Figure 2. The application program is interpreted to be the client or client application. "An application program will typically need to register a listener for each object for which it requires notification" (Col 4, Lines 56-58). Each "application program 23 may generate object instantiation requests for provision to the operating system 22" (Col 6, Lines 19-21). The application program 23 "can establish a listener, generally identified by reference numeral 26, which monitors the respective object 24" (Col 6, Lines 35-37). The listener identifies an event that may occur in the future. The one or more event managers are met by the

listener 26 as shown Figure 2. "Each listener 26 monitors the event information from the object for which it was established, and when it receives an event information item from the object 24 which it (that is, the listener 26) is monitoring, will provide an event notification to the application program 23 which established the listener 26" (Col 6, Lines 51-56). The one or more action handlers are met by application programs 23 as shown in Figure 2. As disclosed, the application programs 23 "can perform predetermined operations" (Col 1, Line 50). The predetermined operations of the application program 23 are in response to notification of the occurrence of an event. The "event manager" or listener 26, in response to detecting an event, is configured to notify the "event broker" or operating system 22 of a first event, which corresponds to the event booking. The application programs are operable to perform specific actions. Subsequently, the "action handler" or application program is notified of the occurrence of the event. A "notification of the event is entered into an event queue maintained by the digital computer system's operating system. The event notifications are generally listed in the event queue in the order in which the operator performs events in connection with the various objects. An application program which needs to be notified of events which occur in connection with an object registers a "listener" for the object, which serve to check the event queue to determine when the event queue includes an event notification for that object" (Col 4, Lines 47-56). The Korn reference fails to explicitly disclose, "each of said event broker, said clients, said event managers, and said action handlers comprise distinct functional entities". However, the examiner takes OFFICIAL NOTICE that it is notoriously well known in the art to use modular

programming techniques so as to hide implementation details and provide a clearly defined interface. Consequently, it would have been obvious to one of ordinary skill in the art to modify Korn with modular programming techniques so as to hide implementation details and provide a clearly defined interface.

In regard to claim 2, Korn et al. discloses an event notification system. The reference fails to explicitly disclose that the event manager is not configured to communicate directly with the first action handler. However, it is submitted that it would have been clearly obvious to one of ordinary skill in the art to have the event manager not configured to communicate directly with the first action handler so as to have an intermediate step between the two modules such as a buffering device so as to alleviate any problems associated with real time communication.

In regard to claim 4, the "event broker" or operating system identifies and selects a first "event manager" or listener 26 from a plurality of listeners. The "event manager" or listener 26, in response to detecting an event, is configured to notify the "event broker" or operating system 22 of a first event, which corresponds to the event booking. "Each listener 26 monitors the event information from the object for which it was established, and when it receives an event information item from the object 24 which it (that is. the listener 26) is monitoring, will provide an event notification to the application program 23 which established the listener 26" (Col 6, Lines 51-56). A "notification of the event is entered into an event queue maintained by the digital computer system's operating system. The event notifications are generally listed in the event queue in the order in which the operator performs events in connection with the various objects. An

application program which needs to be notified of events which occur in connection with an object registers a "listener" for the object, which serve to check the event queue to determine when the event queue includes an event notification for that object" (Col 4, Lines 47-56). As disclosed, the application programs 23 "can perform predetermined operations" (Col 1, Line 50). The first operation of an application program 23 is in response to a notification of the occurrence of a first event.

In regard to claim 7, the "event broker" or operating system 22 is configured to maintain event booking from a plurality of distinct clients. The system comprises "one or more application programs, generally identified by reference numeral 23", where the application program is interpreted to be the client. The one or more application programs are stored on the system. "The processor module 11 includes, for example, processor, memory and mass storage devices such as disk and/or tape storage elements (not separately shown) which perform processing and storage operations in connection with digital data provided thereto. The mass storage subsystems may include such devices as disk or tape subsystems, optical disk storage devices and CD-ROM devices in which information may be stored and/or from which information may be retrieved" (Col 3, Lines 36-44).

In regard to claim 8, the reference discloses that the event bookings comprise a plurality of distinct event types. "Various types of events will be apparent to those skilled in the art" (Col 14, Lines 44-45).

In regard to claim 9, the reference discloses that the action handler "can perform predetermined operations" (Col 1, Line 50). The predetermined operations of the action

handler are in response to notification of the occurrence of an event. It is inherent that the action handler performs a plurality of distinct action types.

In regard to claim 10, Korn et al. discloses an event notification system. The reference fails to explicitly disclose that the event manager is configured to determine whether the booking will be accepted and provide a positive acknowledgement to the broker in response to determining that the event booking is accepted. However, the examiner gives OFFICIAL NOTICE that it is notoriously well known in the art to use the concept of providing positive acknowledgement in response to determining that the entering of a task is accepted so as to ensure that the process performed by the system is performed. Consequently, it would have been clearly obvious to one of ordinary skill in the art to modify Korn et al. with the concept of providing positive acknowledgement in response to determining that the entering of a task is accepted so as to ensure that the process performed by the system is performed.

In regard to claim 12, the event broker is configured to maintain ranks corresponding to the event booking in the form of an event registration list 31, as shown in Figure 2. "The event monitor registration list 31 includes one or more entries 31(1) through 31(M) (generally identified by reference numeral 31(m)). Each entry 31(m) is associated with one of the assistive technology application program 32 which has registered with the event monitor 30. Each entry 31(m) includes a plurality of fields, including a registered application program identifier field 33, and one or more object type identifier fields 34(1) through 34(O) (generally identified by reference numeral 34(o))" (Col 7, Lines 33-41).

In regard to claim 13, Korn et al. discloses an event notification system. The reference discloses that the event broker is in control of accessing the bookings. The event broker is operable to terminate a stored booking as needed. "At some points in their processing, an application program 23 may determine that ones of the graphical user interface objects 24 which were instantiated for it are no longer needed. When that occurs, the application program 23 can so notify the operating system 22, which can terminate the respective object 24. In addition, the application program 23 can terminate the listener 26 which was established to monitor event notification items generated by the object" (Col 6, Lines 57-65). The reference fails to explicitly disclose that access to the bookings booking is based on permissions associated with the accesses and booking. However it is submitted that it would have been clearly obvious to one of ordinary skill in the art to have the access to a booking or task base on permissions so as to prevent the erroneous canceling of a booking.

In regard to claim 14, Korn et al. discloses an event notification system. The one or more application programs are stored on the system. "The processor module 11 includes, for example, processor, memory and mass storage devices such as disk and/or tape storage elements (not separately shown) which perform processing and storage operations in connection with digital data provided thereto. The mass storage subsystems may include such devices as disk or tape subsystems, optical disk storage devices and CD-ROM devices in which information may be stored and/or from which information may be retrieved" (Col 3, Lines 36-44). The reference fails to disclose that the booking has an expiration time. However, the examiner gives OFFICIAL NOTICE

that it is notoriously well known in the art to have an expiration time associated with a booking or a task so as to prevent system resources from being unnecessarily allocated. Consequently, it would have been clearly obvious to one of ordinary skill in the art to modify Korn et al. with an expiration time associated with a booking or a task so as to prevent system resources from being unnecessarily allocated.

In regard to claim 15, the claimed step of receiving a request for an event booking at the event broker, wherein the request identifies a first event and a first action to be taken upon occurrence of the first event is met by the method performed by operating system 22 as shown in Figure 2. The application program is interpreted to be the client or client application. "An application program will typically need to register a listener for each object for which it requires notification" (Col 4, Lines 56-58). Each "application program 23 may generate object instantiation requests for provision to the operating system 22" (Col 6, Lines 19-21). The application program 23 "can establish a listener, generally identified by reference numeral 26, which monitors the respective object 24" (Col 6, Lines 35-37). The claimed step of registering the event-booking request, wherein the event broker notifies a first event manager corresponding to the first event, and notifies a first action handler corresponding to the first action is met by the method performed by the operating system 22 in conjunction with listener 26 as shown in Figure 2. The "event broker" or operating system 22 identifies and selects a first "event manager" or listener 26 from a plurality of listeners. The listener identifies an event that may occur in the future. The "event manager" or listener 26, in response to detecting an event, is configured to notify the "event broker" or operating system 22 of a

first event, which corresponds to the event booking. "Each listener 26 monitors the event information from the object for which it was established, and when it receives an event information item from the object 24 which it (that is. the listener 26) is monitoring, will provide an event notification to the application program 23 which established the listener 26" (Col 6, Lines 51-56). A "notification of the event is entered into an event queue maintained by the digital computer system's operating system. The event notifications are generally listed in the event queue in the order in which the operator performs events in connection with the various objects. An application program which needs to be notified of events which occur in connection with an object registers a "listener" for the object, which serve to check the event queue to determine when the event queue includes an event notification for that object" (Col 4, Lines 47-56). As disclosed, the application programs 23 "can perform predetermined operations" (Col 1, Line 50). The first operation of an application program 23 is in response to a notification of the occurrence of a first event. The claimed step of detecting the first event by the first event manager is met by the method performed by the listener 26 as shown Figure 2. "Each listener 26 monitors the event information from the object for which it was established, and when it receives an event information item from the object 24 which it (that is. the listener 26) is monitoring, will provide an event notification to the application program 23 which established the listener 26" (Col 6, Lines 51-56). The claimed steps of the first event manager notifying the event broker of the detection of the first event, the event broker providing notification to the first action handler of the detection of the first event, and initiating the fist action by the first action handler in response to

detecting the notification from the event broker are met by the operating system 22 in conjunction with listener 26 and application program 23 as shown in Figure 2. The "event manager" or listener 26, in response to detecting an event, is configured to notify the "event broker" or operating system 22 of a first event, which corresponds to the event booking. Subsequently, the "action handler" of application program is notified of the occurrence of the event. A "notification of the event is entered into an event queue maintained by the digital computer system's operating system. The event notifications are generally listed in the event queue in the order in which the operator performs events in connection with the various objects. An application program which needs to be notified of events which occur in connection with an object registers a "listener" for the object, which serve to check the event queue to determine when the event queue includes an event notification for that object" (Col 4, Lines 47-56). The Korn reference fails to explicitly disclose, "each of said event broker, said clients, said event managers, and said action handlers comprise distinct functional entities". However, the examiner takes OFFICIAL NOTICE that it is notoriously well known in the art to use modular programming techniques so as to hide implementation details and provide a clearly defined interface. Consequently, it would have been obvious to one of ordinary skill in the art to modify Korn with modular programming techniques so as to hide implementation details and provide a clearly defined interface.

In regard to claim 16, the application program 23 registers an event with the operating system 22. Operating system 22 constitutes a centralized location. Consequently, the plurality of event bookings are registered in a central location.

In regard to claim 17, the reference discloses that the event bookings comprise a plurality of distinct event types. "Various types of events will be apparent to those skilled in the art" (Col 14, Lines 44-45). The plurality of event managers are met by the listener 26 as shown Figure 2. "Each listener 26 monitors the event information from the object for which it was established, and when it receives an event information item from the object 24 which it (that is. the listener 26) is monitoring, will provide an event notification to the application program 23 which established the listener 26" (Col 6, Lines 51-56).

In regard to claim 18, the step of registering one of the booking events that comprises storing information identifying one of the plurality event managers which is configured to detect the corresponding event is inherent to the method of the disclose system (Col 7, Lines 33-59). The one or more application programs are stored on the system. "The processor module 11 includes, for example, processor, memory and mass storage devices such as disk and/or tape storage elements (not separately shown) which perform processing and storage operations in connection with digital data provided thereto. The mass storage subsystems may include such devices as disk or tape subsystems, optical disk storage devices and CD-ROM devices in which information may be stored and/or from which information may be retrieved" (Col 3, Lines 36-44).

In regard to claim 19, the reference discloses that the action handler "can perform predetermined operations" (Col 1, Line 50). The predetermined operations of

the action handler are in response to notification of the occurrence of an event. It is inherent that the action handler performs a plurality of distinct action types.

In regard to claim 20, the step of registering one of the booking events that comprises storing information identifying one of the plurality action handlers which is configured to initiate the corresponding action is inherent to the method of the disclose system (Col 7, Lines 33-59). The one or more application programs are stored on the system. "The processor module 11 includes, for example, processor, memory and mass storage devices such as disk and/or tape storage elements (not separately shown) which perform processing and storage operations in connection with digital data provided thereto. The mass storage subsystems may include such devices as disk or tape subsystems, optical disk storage devices and CD-ROM devices in which information may be stored and/or from which information may be retrieved" (Col 3, Lines 36-44).

4. Claims 1, 5, 22-25 and 28-29 rejected under 35 U.S.C. 103(a) as being unpatentable over Lawler et al. (US Pat No 5,699,107) in view of Korn et al.

In regard to claim 1, Lawler et al. discloses a program reminder system that reminds a user of an interactive viewing system when a pre-selected program is available. The reference is silent with respect to a first event manager, a first action handler, and an event broker mechanism. Korn et al. teaches an event notification system. The event broker is met by the operating system 22 as shown in Figure 2. The application program is interpreted to be the client or client application. "An application program will typically need to register a listener for each object for which it requires

notification" (Col 4, Lines 56-58). Each "application program 23 may generate object instantiation requests for provision to the operating system 22" (Col 6, Lines 19-21). The application program 23 "can establish a listener, generally identified by reference numeral 26, which monitors the respective object 24" (Col 6, Lines 35-37). The one or more event managers are met by the listener 26 as shown Figure 2. "Each listener 26 monitors the event information from the object for which it was established, and when it receives an event information item from the object 24 which it (that is. the listener 26) is monitoring, will provide an event notification to the application program 23 which established the listener 26" (Col 6, Lines 51-56). The one or more action handlers are met by application programs 23 as shown in Figure 2. As disclosed, the application programs 23 "can perform predetermined operations" (Col 1, Line 50). The predetermined operations of the application program 23 are in response to notification of the occurrence of an event. The "event manager" or listener 26, in response to detecting an event, is configured to notify the "event broker" or operating system 22 of a first event, which corresponds to the event booking. Subsequently, the "action handler" or application program is notified of the occurrence of the event. A "notification of the event is entered into an event queue maintained by the digital computer system's operating system. The event notifications are generally listed in the event queue in the order in which the operator performs events in connection with the various objects. An application program which needs to be notified of events which occur in connection with an object registers a "listener" for the object, which serve to check the event queue to determine when the event queue includes an event notification for that object" (Col 4,

Lines 47-56). Korn et al. teaches this event notification system so as to provide a simplified system and method of providing notification of events. Consequently, it would have been clearly obvious to one of ordinary skill in the art to modify Lawler et al. with the Korn et al. event notification system for the stated advantage. The combination of Korn and Lawler fail to explicitly disclose, "each of said event broker, said clients, said event managers, and said action handlers comprise distinct functional entities". However, the examiner takes OFFICIAL NOTICE that it is notoriously well known in the art to use modular programming techniques so as to hide implementation details and provide a clearly defined interface. Consequently, it would have been obvious to one of ordinary skill in the art to modify the combined teaching with modular programming techniques so as to hide implementation details and provide a clearly defined interface.

In regard to claim 5, Lawler et al. discloses a system to remind a use of an event (i.e. the start of a television program). (Col 3, Lines 36-50).

In regard to claim 22, Lawler et al. discloses a program reminder system that reminds a user of an interactive viewing system when a pre-selected program is available. The reference discloses a signal receiver configured to receive a broadcast signal as shown in Figure 2 (Items 48, 50 and 66). The reference is silent with respect to a first event manager, a first action handler, and an event broker mechanism. Korn et al. teaches an event notification system. The event broker is met by the operating system 22 as shown in Figure 2. The application program is interpreted to be the client or client application. "An application program will typically need to register a listener for each object for which it requires notification" (Col 4, Lines 56-58). Each "application

program 23 may generate object instantiation requests for provision to the operating system 22" (Col 6, Lines 19-21). The application program 23 "can establish a listener, generally identified by reference numeral 26, which monitors the respective object 24" (Col 6, Lines 35-37). The listener identifies an event that may occur in the future. The one or more event managers are met by the listener 26 as shown Figure 2. "Each listener 26 monitors the event information from the object for which it was established, and when it receives an event information item from the object 24 which it (that is. the listener 26) is monitoring, will provide an event notification to the application program 23 which established the listener 26" (Col 6, Lines 51-56). The one or more action handlers are met by application programs 23 as shown in Figure 2. As disclosed, the application programs 23 "can perform predetermined operations" (Col 1, Line 50). The predetermined operations of the application program 23 are in response to notification of the occurrence of an event. The "event manager" or listener 26, in response to detecting an event, is configured to notify the "event broker" or operating system 22 of a first event, which corresponds to the event booking. Subsequently, the "action handler" or application program is notified of the occurrence of the event. A "notification of the event is entered into an event queue maintained by the digital computer system's operating system. The event notifications are generally listed in the event queue in the order in which the operator performs events in connection with the various objects. An application program which needs to be notified of events which occur in connection with an object registers a "listener" for the object, which serve to check the event queue to determine when the event queue includes an event notification for that object" (Col 4,

Lines 47-56). Korn et al. teaches this event notification system so as to provide a simplified system and method of providing notification of events. Consequently, it would have been clearly obvious to one of ordinary skill in the art to modify Lawler et al. with the Korn et al. event notification system for the stated advantage. The combination of Korn and Lawler fail to explicitly disclose, "each of said event broker, said clients, said event managers, and said action handlers comprise distinct functional entities". However, the examiner takes OFFICIAL NOTICE that it is notoriously well known in the art to use modular programming techniques so as to hide implementation details and provide a clearly defined interface. Consequently, it would have been obvious to one of ordinary skill in the art to modify the combined teaching with modular programming techniques so as to hide implementation details and provide a clearly defined interface.

In regard to claim 23, Lawler et al. discloses an interactive television system receiver coupled to receive a broadcast signal and configured to provide an output signal to a television. "The demodulator 52 functions as a conventional television tuner for selecting one or more of multiple conventional analog video signals present at input 48" (Col 6, Lines 12-15). "The interactive station controller 18 also may include a graphics subsystem 62 that is controlled by the CPU 58 to form graphics images, including user interface displays, on the video display 20" (Col 6, Lines 32-35).

In regard to claim 24, Korn et al. discloses that the "event broker" or operating system identifies and selects a first "event manager" or listener 26 from a plurality of listeners. The "event manager" or listener 26, in response to detecting an event, is configured to notify the "event broker" or operating system 22 of a first event, which

corresponds to the event booking. "Each listener 26 monitors the event information from the object for which it was established, and when it receives an event information item from the object 24 which it (that is. the listener 26) is monitoring, will provide an event notification to the application program 23 which established the listener 26" (Col 6, Lines 51-56). A "notification of the event is entered into an event queue maintained by the digital computer system's operating system. The event notifications are generally listed in the event queue in the order in which the operator performs events in connection with the various objects. An application program which needs to be notified of events which occur in connection with an object registers a "listener" for the object, which serve to check the event queue to determine when the event queue includes an event notification for that object" (Col 4, Lines 47-56). Furthermore, "The object type identifier portion 35(o) of each object type identifier field 34(o) contains an object type identifier which identifies one type of graphical user interface object 24 for which the assistive technology application program 32 for which the entry 31(m) was established, and the event type list portion 36(o) of the same object type identifier field 34(o) contains the identification of the type or types of events which occur in connection with the object(s) of the type identified in the object type identifier portion 35(o), for which the assistive technology application program 32 is to be notified" (Col 7, Lines 44-54). As disclosed, the application programs 23 "can perform predetermined operations" (Col 1, Line 50). The first operation of an application program 23 is in response to a notification of the occurrence of a first event.

In regard to claim 25, Korn et al. discloses an event notification system. The reference fails to explicitly disclose that the event manager is not configured to communicate directly with the first action handler. However, it is submitted that it would have been clearly obvious to one of ordinary skill in the art to have the event manager not configured to communicate directly with the first action handler so as to have an intermediate step between the two modules such as a buffering device so as to alleviate any problems associated with real time communication.

In regard to claim 28, Korn et al. discloses an event notification system. The reference discloses that the event broker is in control of accessing the bookings. The event broker is operable to terminate a stored booking as needed. "At some points in their processing, an application program 23 may determine that ones of the graphical user interface objects 24 which were instantiated for it are no longer needed. When that occurs, the application program 23 can so notify the operating system 22, which can terminate the respective object 24. In addition, the application program 23 can terminate the listener 26 which was established to monitor event notification items generated by the object" (Col 6, Lines 57-65). The reference fails to explicitly disclose that access to the bookings booking is based on permissions associated with the accesses and booking. However it is submitted that it would have been clearly obvious to one of ordinary skill in the art to have the access to a booking or task base on permissions so as to prevent the erroneous canceling of a booking.

In regard to claim 29, the one or more application programs are stored on the system. "The processor module 11 includes, for example, processor, memory and

mass storage devices such as disk and/or tape storage elements (not separately shown) which perform processing and storage operations in connection with digital data provided thereto. The mass storage subsystems may include such devices as disk or tape subsystems, optical disk storage devices and CD-ROM devices in which information may be stored and/or from which information may be retrieved" (Col 3, Lines 36-44).

5. Claims 3 and 6 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lawler et al. in view of Korn et al and in further view of Chawla (US Pat No 6,108,695).

In regard to claim 3, the combination of Lawler et al. and Korn et al. teaches an event notification system implemented in a receiver for a broadcast system. The combination of references fails to explicitly disclose the use of a framework comprising a software layer between an application layer and a driver layer. Chawla teaches the use of a software layer between the "application layer" or the media stream manager and the "driver layer" or the low-level software (Figure 3; Col 4, Lines 6-12), which is preferred in order to increase system performance and user control. Consequently, it would have been clearly obvious to one of ordinary skill in the art to modify the combination of references with a software layer between the "application layer" or the media stream manager and the "driver layer" or the low-level software to increase system performance and user control.

In regard to claim 6, Chawla discloses a system for managing channels on a multiple channel digital media server. The reference fails to explicitly disclose the use of library extensions as claimed. However, the examiner gives OFFICIAL NOTICE that

it is notoriously well known in the art to use library extensions for addressing files.

Consequently, it would have been clearly obvious to one of ordinary skill in the art to implement Chawla with library extensions for addressing files.

6. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Korn et al in view of Sudhakaran et al. (US Pat No 6,636,901).

In regard to claim 11, Korn et al. discloses an event notification system. The reference fails to explicitly disclose that the event broker is configured to determine the recourses required and resolve resource conflicts. Sudhakaran et al. teaches automatically determining the resources required and resolving any resource conflicts so that one or more resources can be shared between different functions in an error free fashion. Consequently, it would have been clearly obvious to one of ordinary skill in the art to modify Korn et al. with automatically determining the resources required and resolving any resource conflicts so that one or more resources can be shared between different functions in an error free fashion.

7. Claims 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Korn et al. in view of Chernock et al. (US Pat Application Publication No 2003/0159150).

In regard to claim 21, Korn et al. discloses an event notification system. The reference fails to explicitly disclose actions as claimed. The reference fails to explicitly disclose a plurality of distinct event types and actions. Chernock teaches the use of plurality of distinct event types and actions. Some of these event types and actions are:

- “Tuning the receiving device to play the subsequent program”

- “Recording a subsequent program or its selected embedded content at the scheduled time on an external video recording medium for video, on an external audio recording medium for audio, and on internal or external digital data recording medium for other digital data”
- “Reminding the user of the scheduled event at the scheduled time, with a video or audio notification, which will allow the user to tune the STB device to play the program” (Paragraphs 0018-0020).

This system is preferred in order to provide the user with more options. Consequently, it would have been clearly obvious to one of ordinary skill in the art to implement Korn et al. with of distinct event types to provide the users with more options.

8. Claim 27 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lawler et al. in view of Korn et al. and in further view of Sudhakaran et al.

In regard to claim 27, the combination of Lawler et al. and Korn et al. disclose that multiple booking may be register. The combination fails to explicitly disclose that the event broker is configured to determine the recourses required and resolve resource conflicts. Sudhakaran et al. teaches automatically determining the resources required and resolving any resource conflicts so that one or more resources can be shared between different functions in an error free fashion. Consequently, it would have been clearly obvious to one of ordinary skill in the art to modify the combination of Lawler et al. and Korn et al. with automatically determining the resources required and resolving any resource conflicts so that one or more resources can be shared between different functions in an error free fashion.

***Allowable Subject Matter***

9. Claim 26 objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

***Conclusion***

10. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure is as follows:

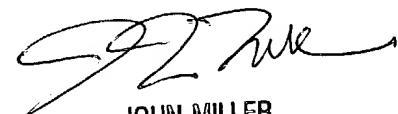
- The Cohen et al. (US Pat No 6,782,541) discloses a system and method of exchanging information between software modules.
- The Page et al. (US Pat No 5,812,768) discloses a System for allocating adaptor to server by determining from embedded foreign protocol commands in client request if the adapter service matches the foreign protocol.

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to John Manning whose telephone number is 703-305-0345. The examiner can normally be reached on M-F: 8:00 - 5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John W Miller can be reached on 703-305-4795. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JM  
October 14, 2004



JOHN MILLER  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2600